JUNOS

Class of Service Using IPv6 DiffServ Feature Guide

Release 9.5

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Part Number: 530-028963-01, Revision 1

JUNOS Class of Service Using Ipv6 DiffServ,
Release 9.5

Writing: Fawn Damitio, Richard Hendricks, and Walter Goralski
Editing: Sonia Saruba
Illustration: Faith Bradford, Fawn Damitio, Nathaniel Woodward, and Richard Hendricks
Cover Design: Edmonds Design

Revision History
13 April 2009–530-028963-01—Revision 1

The information in this document is current as of the date listed in the revision history.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. The JUNOS software has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE. BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMER OR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. **The Parties.** The parties to this Agreement are (i) Juniper Networks, Inc. (if the Customer's principal office is located in the Americas) or Juniper Networks (Cayman) Limited (if the Customer's principal office is located outside the Americas) (such applicable entity being referred to herein as "Juniper"), and (ii) the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. **The Software.** In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller, or which was embedded by Juniper in equipment which Customer purchased from Juniper or an authorized Juniper reseller. "Software" also includes updates, upgrades and new releases of such software. "Embedded Software" means Software which Juniper has embedded in or loaded onto the Juniper equipment and any updates, upgrades, additions or replacements which are subsequently embedded in or loaded onto the equipment.

3. **License Grant.** Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:

a. Customer shall use Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.

b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations are physically contained on a single chassis.

c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.

d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.

e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. **Use Prohibitions.** Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use Embedded Software on non-Juniper equipment; (j) use Embedded Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

5. **Audit.** Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. **Confidentiality.** The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software for Customer's internal business purposes.

7. **Ownership.** Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. **Warranty, Limitation of Liability, Disclaimer of Warranty.** The warranty applicable to the Software shall be as set forth in the warranty statement that accompanies the Software (the "Warranty Statement"). Nothing in this Agreement shall give rise to any obligation to support the Software. Support services may be purchased separately. Any such support shall be governed by a separate, written support services agreement. TO THE MAXIMUM EXTENT PERMITTED BY LAW, JUNIPER SHALL NOT BE LIABLE FOR ANY LOST PROFITS, LOSS OF DATA, OR COSTS OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, THE SOFTWARE, OR ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. IN NO EVENT SHALL JUNIPER BE LIABLE FOR DAMAGES ARISING FROM UNAUTHORIZED OR IMPROPER USE OF ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. EXCEPT AS EXPRESSLY PROVIDED IN THE WARRANTY STATEMENT TO THE EXTENT PERMITTED BY LAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE), INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

9. **Termination.** Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. **Taxes.** All license fees payable under this agreement are exclusive of tax. Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software. If applicable, valid exemption documentation for each taxing jurisdiction shall be provided to Juniper prior to invoicing, and Customer shall promptly notify Juniper if their exemption is revoked or modified. All payments made by Customer shall be net of any applicable withholding tax. Customer will provide reasonable assistance to Juniper in connection with such withholding taxes by promptly: providing Juniper with valid tax receipts and other required documentation showing Customer's payment of any withholding taxes; completing appropriate applications that would reduce the amount of withholding tax to be paid; and notifying and assisting Juniper in any audit or tax proceeding related to transactions hereunder. Customer shall comply with all applicable tax laws and regulations, and Customer will promptly pay or reimburse Juniper for all costs and damages related to any liability incurred by Juniper as a result of Customer's non-compliance or delay with its responsibilities herein. Customer's obligations under this Section shall survive termination or expiration of this Agreement.

11. **Export.** Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to Customer may contain encryption or other capabilities restricting Customer's ability to export the Software without an export license.

12. **Commercial Computer Software.** The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4, FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14(ALT III) as applicable.

13. **Interface Information.** To the extent required by applicable law, and at Customer's written request, Juniper shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of applicable fee, if any. Customer shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which Juniper makes such information available.

14. **Third Party Software.** Any licensor of Juniper whose software is embedded in the Software and any supplier of Juniper whose products or technology are embedded in (or services are accessed by) the Software shall be a third party beneficiary with respect to this Agreement, and such licensor or vendor shall have the right to enforce this Agreement in its own name as if it were Juniper. In addition, certain third party software may be provided with the Software and is subject to the accompanying license(s), if any, of its respective owner(s). To the extent portions of the Software are distributed under and subject to open source licenses obligating Juniper to make the source code for such portions publicly available (such as the GNU General Public License ("GPL") or the GNU Library General Public License ("LGPL")), Juniper will make such source code portions (including Juniper modifications, as appropriate) available upon request for a period of up to three years from the date of distribution. Such request can be made in writing to Juniper Networks, Inc., 1194 N. Mathilda Ave., Sunnyvale, CA 94089, ATTN: General Counsel. You may obtain a copy of the GPL at http://www.gnu.org/licenses/gpl.html, and a copy of the LGPL at http://www.gnu.org/licenses/lgpl.html.

15. **Miscellaneous.** This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous

agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentés confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattaché, soient redigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Table of Contents

List of Figures

List of Tables

Part 1 Class of Service Using IPv6 DiffServ

Part 1
Class of Service Using IPv6 DiffServ

Chapter 1

Class of Service Using IPv6 DiffServ Concepts and Reference Material

Overview of Class of Service Using IPv6 DiffServ

Class of service (CoS) is the assignment of traffic flows to different service levels. Service providers can use router-based CoS features to define service levels that provide different delay, jitter (delay variation), and packet loss characteristics to particular applications served by specific traffic flows.

Usually, IP routers forward packets independently and without any control on throughput or delay. This is known as *best-effort* service. This service is as good as the network equipment and links, and the result is satisfactory for many traditional IP applications emphasizing data delivery, such as e-mail or Web browsing. However, newer IP applications such as real-time video and audio (or voice) require lower delay, jitter, and loss parameters than simple best-effort networks can provide. CoS is intended for networks supporting these types of time-sensitive video and audio applications.

A router cannot compromise best-effort forwarding performance in order to deliver CoS features, because this merely trades one problem for another. When CoS features are enabled, they must allow routers to better process critical packets as well as best-effort traffic flows, even during times of congestion. Network throughput is determined by a combination of available bandwidth and delay. CoS guarantees a minimum bandwidth dedicated to a service class.

The main impact of CoS on network delay is in queueing delays, when packets are normally queued for output in the order of arrival, regardless of service class. Queuing delays increase with network congestion and often result in lost packets when queue buffers overflow. The other two elements of overall network delay, serial transmission delays determined by link speeds and propagation delays determined by media type, are not affected by CoS settings.

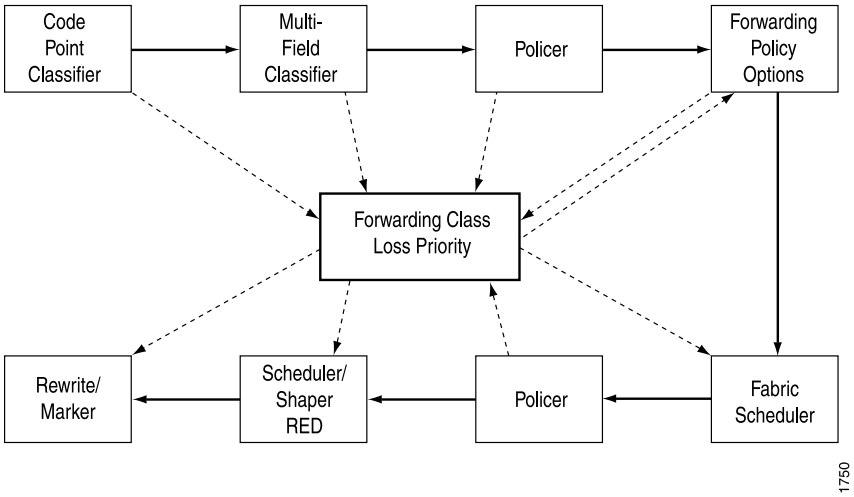
Any CoS implementation must work consistently end to end through the network. A standards-based, vendor-neutral CoS implementation satisfies this requirement best. Juniper Networks CoS features interoperate with other vendors' CoS implementations because they are based on IETF Differentiated Services (DiffServ) standards.

DiffServ specifications establish a six-bit field in the IPv4 and IPv6 packet header to indicate the service class that should be applied to the packet. The bit values in the DiffServ field form DiffServ code points (DSCPs) that can be set by the application or a router on the edge of a DiffServ-enabled network.

Although CoS methods such as DiffServ specify the position and length of the DSCP in the packet header, the implementation of the router mechanisms to deliver DiffServ internally is vendor-specific. CoS functions in JUNOS software are configured through a series of mechanisms that you can configure individually or in combination to define particular service offerings.

Figure 1 on page 4 shows the components of the JUNOS CoS features, illustrating the sequence in which they interact.

Figure 1: Packet Flow Through CoS-Configurable Components



You can configure one or more of the following JUNOS CoS mechanisms:

- Classifiers—Allow you to associate incoming packets with a forwarding class and packet loss priority (PLP). Two general types of classifiers are supported:
  - Behavior aggregate (BA) or code point traffic classifiers allow you to set the forwarding class and PLP based on DSCP.
  - Multifield (MF) traffic classifiers allow you to set the forwarding class and PLP based on firewall filter rules. This is usually done at the edge of the network for packets that do not have valid DSCPs in the packet headers.
- Forwarding classes—Allow you to set the scheduling and marking of packets as they transit the router. Known as ordered aggregates in the DiffServ architecture, the forwarding class plus the loss priority determine the router's per-hop behavior (PHB in DiffServ) for CoS.

- Loss priorities—Allow you to set the priority of dropping a packet before it is sent. Loss priority affects the scheduling of a packet without affecting the packet's relative ordering.
- Forwarding policy options—Allow you to associate forwarding classes with next hops. Forwarding policy options also allow you to create classification overrides, which assign forwarding classes to sets of prefixes.
- Transmission scheduling and rate control—Provide you with a variety of tools to manage traffic flows:
  - Schedulers—Allow you to define the priority, bandwidth, delay buffer size, rate control status, and RED drop profiles to be applied to a particular forwarding class for packet transmission.
  - Fabric schedulers—For M120, M320, and T-series routing platforms only, fabric schedulers allow you to identify a packet as high or low priority based on its forwarding class, and to associate schedulers with the fabric priorities.
  - Policers for traffic classes—Allow you to limit traffic of a certain class to a specified bandwidth and burst size. Packets exceeding the policer limits can be discarded, or can be assigned to a different forwarding class or to a different loss priority, or to both. You define policers with filters that can be associated with input or output interfaces.
- Rewrite markers—Allow you to redefine the DSCP value of outgoing packets. Rewriting or marking outbound packets is useful when the routing platform is at the border of a network and must alter the code points to meet the policies of the targeted peer.

Typically, rewrites of the DSCPs on outgoing packets are done once, when packets enter the DiffServ portion of the network, either because the packets do not arrive from the customer with the proper DSCP bit set or because the service provider wishes to verify that the customer has set the DSCP properly. CoS schemes that accept the DSCP and classify and schedule traffic solely on DSCP value perform behavior aggregate (BA) DiffServ functions and do not usually rewrite the DSCP. DSCP rewrites typically occur in multifield (MF) DiffServ scenarios.

Related Topics

- Class of Service Using IPv6 Solutions Page
- Roadmap for Configuring CoS with IPv6 DiffServ on page 13
- System Requirements for CoS with DiffServ for IPv6 on page 12
- Example: CoS with IPv6 DiffServ Configuration on page 24

Default DHCP Mappings

Table 1 on page 6 shows the mapping of DiffServ service class meanings (aliases) to DSCPs.

Table 1: Default DSCP Mappings

DiffServ Service Class Alias	IPv4 and IPv6 DSCP Mapping
ef	101110
af11	001010
af12	001100
af13	001110
af21	010010
af22	010100
af23	010110
af31	011010
af32	011100
af33	011110
af41	100010
af42	100100
af43	100110
be	000000
cs1	001000
cs2	010000
cs3	011000
cs4	100000
cs5	101000
nc1/cs6	110000
nc2/cs7	111000

None of the aliases are established by DiffServ specifications. The aliases are well-known only through usage. For example, it is widely accepted that the alias for DSCP 101110 is ef (expedited forwarding). The 21 well-known DSCPs establish 5 DiffServ service classes:

- **Best-effort (be)**—The router does not apply any special CoS handling to packets with 000000 in the DiffServ field, a backward compatibility feature. There is usually a high probability that these packets will be dropped under congested network conditions.
- **Assured forwarding (af)**—The router offers a high level of assurance that the packets are delivered as long as the packet flow from the customer stays within a certain service profile (the service provider defines the values). The router accepts excess traffic, but applies a random early discard (RED) drop profile to decide if the excess packets should be dropped and not forwarded. Three drop probabilities (low, medium, and high) are defined for this service class.
- **Expedited forwarding (ef)**—The router delivers assured bandwidth, low loss, low delay, and low delay variation (jitter) end-to-end for packets in this service class. Routers accept excess traffic in this class, but in contrast to assured forwarding, out-of-profile expedited-forwarding packets can be forwarded out of sequence or dropped.
- **Conversational services (cs)**—The router delivers assured (usually low) bandwidth with low delay and jitter for packets in this service class. Packets can be dropped, but never delivered out of sequence. Packetized voice is a good example of a conversational service.
- **Network control (nc)**—The router delivers packets in this service class with a low priority (these packets are not delay-sensitive). Typically, these packets represent routing protocol hello or keepalive messages and loss of these packets jeopardizes proper network operation, so delay is preferable to discard.

Related Topics

- Class of Service Using IPv6 Solutions Page
- Overview of Class of Service Using IPv6 DiffServ on page 3
- System Requirements for CoS with DiffServ for IPv6 on page 12
- Roadmap for Configuring CoS with IPv6 DiffServ on page 13
- Example: CoS with IPv6 DiffServ Configuration on page 24

Default Forwarding Classes

Table 2 on page 8 shows the default forwarding class and packet loss priority (PLP) for the well-known DSCPs. It is important to note that although several DSCPs map to the expedited-forwarding and assured-forwarding classes, by default no resources are assigned to these forwarding classes. All of these settings can be changed through configuration.

Table 2: Default Behavior Aggregate Classification

DSCP and DSCP IPv6	Forwarding Class	PLP
ef	expedited-forwarding	low
af11	assured-forwarding	low
af12	assured forwarding	high
af13	assured forwarding	high
af21	best-effort	low
af22	best-effort	low
af23	best-effort	low
af31	best-effort	low
af32	best-effort	low
af33	best-effort	low
af41	best-effort	low
af42	best-effort	low
af43	best-effort	low
be	best-effort	low
cs1	best-effort	low
cs2	best-effort	low
cs3	best-effort	low
cs4	best-effort	low
cs5	best-effort	low
nc1/cs6	network-control	low
nc2/cs7	network control	low
other	best-effort	low

Table 3 on page 9 shows the router forwarding classes associated with each well-known DSCP code point and the resources assigned to their output queues.

Table 3: Classification Forwarding Classes and Queues

DCSP Alias	DSCP Bits	Forwarding Class	PLP	Queue
ef	101110	expedited-forwarding	low	1
af11	001010	assured-forwarding	low	2
af12	001100	assured-forwarding	high	2
af13	001110	assured-forwarding	high	2
af21	010010	best-effort	low	0
af22	010100	best-effort	low	0
af23	010110	best-effort	low	0
af31	011010	best-effort	low	0
af32	011100	best-effort	low	0
af33	011110	best-effort	low	0
af41	100010	best-effort	low	0
af42	100100	best-effort	low	0
af43	100110	best-effort	low	0
be	000000	best-effort	low	0
cs1	001000	best-effort	low	0
cs2	010000	best-effort	low	0
cs3	011000	best-effort	low	0
cs4	100000	best-effort	low	0
cs5	101000	best-effort	low	0
nc1/cs6	110000	network-control	low	3
nc2/cs7	111000	network-control	low	3
other	—	best-effort	low	0

Table 4 on page 10 shows the resources assigned to the four forwarding classes in this example.

Table 4: Resources Assigned to Queues

Queue	Forwarding Class	Transmit Rate	Buffer Size	Priority
0	be (data)	40 %	40 %	Low
1	ef (financial)	10 %	10 %	High
2	af (audiovisual)	45 %	45 %	High (with RED)
3	nc (network control)	5 %	5 %	Low

The table shows how the 95 percent of output link transmission rate and buffer size (queue) resources assigned by default to Q0 (best-effort) are distributed to Q1 (expedited forwarding) and Q2 (assured forwarding). The audiovisual traffic consumes more bandwidth than other applications, but the financial information, although critical, is carried in fewer packets. In keeping with DiffServ specifications, a RED drop profile is applied to the assured forwarding class. The financial data has a strict set of traffic parameters that must be respected.

The three DiffServ assured forwarding classes supported (af11, af12, and af13, with low, medium, and high packet drop probability, respectively) are distinguished by using a low PLP and RED drop profile for af11 and a high PLP and RED for af12 and af13. All of these parameters should be closely monitored initially for performance and adjusted as necessary.

Related Topics

- Class of Service Using IPv6 Solutions Page
- Overview of Class of Service Using IPv6 DiffServ on page 3
- System Requirements for CoS with DiffServ for IPv6 on page 12
- Roadmap for Configuring CoS with IPv6 DiffServ on page 13
- Example: CoS with IPv6 DiffServ Configuration on page 24

Juniper Networks Default Forwarding Classes

Most M-series routers have only four queues built into the hardware. M120, M320, MX-series, and T-series routing platforms can be configured for up to eight queues. If a classifier does not assign a packet to any other queue (for example, for other than well-known DSCPs that have not been added to the classifier), the packet is assigned by default to the class associated with queue 0 (Q0).

Table 5 on page 11 shows the four forwarding classes and queues to which Juniper Networks classifiers assign a packet based on the DSCP values in arriving packet headers.

Table 5: Default Forwarding Classes

Forwarding Class Name	Queue
best-effort	queue 0
expedited-forwarding	queue 1
assured-forwarding	queue 2
network-control	queue 3

Each forwarding class has an associated scheduler priority. Only two forwarding classes, best-effort and network-control (Q0 and Q3), are actually referenced in the default scheduler configuration. However, you can manually configure resources for the expedited-forwarding and assured-forwarding classes (Q1 and Q2).

The default scheduler settings are not visible in the output of the show class-of-service command; rather, they are implicit.

Default Scheduler

```
[edit class-of-service]
schedulers {
    network-control {
        transmit-rate percent 5;
        buffer-size percent 5;
        priority low;
        drop-profile-map loss-priority any protocol any;
        drop-profile terminal;
    }
    best-effort {
        transmit-rate percent 95;
        buffer-size percent 95;
        priority low;
        drop-profile-map loss-priority any protocol any;
        drop-profile terminal;
    }
}
drop-profiles {
    terminal {
        fill-level 100 drop-probability 100;
    }
}
```

By default, the best-effort forwarding class (Q0) receives 95 percent of the output link bandwidth and buffer space, and the network-control forwarding class (Q3) receives 5 percent of the output link bandwidth and buffer space. The default drop profile provides *tail drop*, where the buffer fills and then discards all packets until there is space in the buffer again. There are no schedulers for the expedited-forwarding or assured-forwarding classes because by default no resources are assigned to Q1 and Q2.

All af classes other than af1*x* are mapped to best-effort, since RFC 2597 prohibits a node from aggregating classes. In effect, mapping to best-effort implies that the node does not support that class.

Related Topics

- Class of Service Using IPv6 Solutions Page
- Overview of Class of Service Using IPv6 DiffServ on page 3
- System Requirements for CoS with DiffServ for IPv6 on page 12
- Roadmap for Configuring CoS with IPv6 DiffServ on page 13
- Example: CoS with IPv6 DiffServ Configuration on page 24

System Requirements for CoS with DiffServ for IPv6

To implement CoS with DiffServ for IPv6, your system must meet these minimum requirements:

- JUNOS Release 8.2 or later for MX-series routers
- JUNOS Release 7.2 or later for J-series Services routers
- JUNOS Release 6.3 or later for M-series and T-series routers
- Three Juniper Networks J-series, M-series, MX-series, or T-series routers
- For M-series routers, Enhanced FPCs capable of supporting DSCPs and, for MF classifiers, Internet Processor II ASICs

Related Topics

- Class of Service Using IPv6 Solutions Page
- Overview of Class of Service Using IPv6 DiffServ on page 3
- System Requirements for CoS with DiffServ for IPv6 on page 12
- Roadmap for Configuring CoS with IPv6 DiffServ on page 13
- Example: CoS with IPv6 DiffServ Configuration on page 24

Terms and Acronyms for CoS with DiffServ for IPv6

C

class of service A set of forwarding class parameters that define different treatment for different traffic flows.

classifier A method of reading a sequence of bits in a packet header or label and determining the packet's forwarding class.

D

Differentiated Services (DiffServ) A standards-based method of associating CoS parameters with traffic flows and their forwarding classes.

Differentiated Services code point (DSCP) Values for a 6-bit field defined for IPv4 and IPv6 packet headers that can be used to enforce CoS distinctions in routers.

Chapter 2

Configuring CoS with IPv6 DiffServ

Roadmap for Configuring CoS with IPv6 DiffServ

To configure class of service (CoS) over IPv6, you must:

- Configure a multifield (MF) classifier for IPv6 to detect packets of interest to CoS and assign the packet to the proper forwarding class independently of Differentiated Services Code Point (DSCP). See “Configuring a Firewall Filter for an MF Classifier on Customer Interfaces” on page 14.

  Next, apply the MF classifier to the appropriate interface. See “Applying the Firewall Filter to Customer Interfaces” on page 15.

- Assign the forwarding classes established by the MF classifier to output queues. See “Assigning Forwarding Classes to Output Queues” on page 16.

- Configure rewrite rules to replace DSCPs on packets received from the customer with the values expected by other routers. See “Configuring Rewrite Rules” on page 16.

  Next, apply the rewrite rules to the appropriate interface. See “Applying Rewrite Rules to an Interface” on page 17.

- Configure behavior aggregate (BA) classifiers for IPv6 on network interfaces because the DSCPs have been explicitly rewritten on the edge routers. See “Configuring BA Classifiers” on page 17.

  Next, apply the BA classifier to the appropriate interface. See “Applying a BA Classifier to an Interface” on page 18.
- Configure random early discard (RED) drop profiles to determine the probability of DiffServ assured forwarding packets being discarded under congested conditions. See “Configuring RED Drop Profiles” on page 19.
- Configure schedulers to assign resources, priorities, and drop profiles to output queues. See “Configuring Schedulers” on page 19.
- Configure a scheduler map to assign a forwarding class to a scheduler. See “Configuring Scheduler Maps” on page 20.

  Next, apply the scheduler map to the appropriate interface. See “Applying a Scheduler Map to an Interface” on page 20.

Related Topics

- Class of Service Using IPv6 Solutions Page
- Overview of Class of Service Using IPv6 DiffServ on page 3
- System Requirements for CoS with DiffServ for IPv6 on page 12
- Example: CoS with IPv6 DiffServ Configuration on page 24

Configuring a Firewall Filter for an MF Classifier on Customer Interfaces

You configure an MF classifier for IPv6 to detect packets of interest to CoS and assign the packet to the proper forwarding class independently of DSCP. To configure an MF classifier on a customer-facing link, configure a policer for the expedited forwarding traffic and a firewall filter to classify traffic.

```
[edit firewall]
policer ef-FIN-Policer-Profile {
    if-exceeding {
        bandwidth-percent 10;
        burst-size-limit 2k;
    }
    then loss-priority high;
}
family inet6 {
    filter mf-classifier {
        filter-specific;
        term AV {
            from {
                destination-address {
                    0:0:FFFF:172.16.79.11;
                }
            }
            then {
                loss-priority low;
                forwarding-class af-AV-class;
            }
```

```
        }
        term Finance {
            from {
                destination-address {
                    0:0:FFFF:172.16.79.63;
                }
            }
            then {
                policer ef-FIN-Policer-Profile;
                forwarding-class ef-FIN-class;
            }
        }
        term Network-Control {
            from {
                traffic-class 192; # 192 is the 110000 traffic class.
            }
            then {
                forwarding-class nc-CONTROL-class; # This is network control traffic.
            }
        }
        term Data {
            then forwarding-class be-DATA-class; # The rest is data.
        }
    }
}
```

Related Topics

- Applying the Firewall Filter to Customer Interfaces on page 15
- Class of Service Using IPv6 Solutions Page
- Roadmap for Configuring CoS with IPv6 DiffServ on page 13
- Overview of Class of Service Using IPv6 DiffServ on page 3
- System Requirements for CoS with DiffServ for IPv6 on page 12
- Example: CoS with IPv6 DiffServ Configuration on page 24

Applying the Firewall Filter to Customer Interfaces

You apply an MF classifier firewall filter for IPv6 to customer interfaces. To apply an MF classifier firewall filter on customer-facing links, apply the classifier as an input filter at the [edit interfaces] hierarchy level.

```
[edit interfaces]
so-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.54.1/24;
        }
        family inet6 {
            filter {
                input mf-classifier;
            }
            address 0:0:FFFF:192.168.54.1/120;
        }
```

```
    }
}
```

Related Topics

- Class of Service Using IPv6 Solutions Page
- Configuring a Firewall Filter for an MF Classifier on Customer Interfaces on page 14
- Roadmap for Configuring CoS with IPv6 DiffServ on page 13
- Overview of Class of Service Using IPv6 DiffServ on page 3
- System Requirements for CoS with DiffServ for IPv6 on page 12
- Example: CoS with IPv6 DiffServ Configuration on page 24

Assigning Forwarding Classes to Output Queues

You must assign the forwarding classes established by the MF classifier to output queues. To assign a forwarding class to an output queue, include the forwarding-classes statement at the [edit class-of-service] hierarchy level.

```
[edit class-of-service]
forwarding-classes {
    queue 0 be-DATA-class;
    queue 1 ef-FIN-class;
    queue 2 af-AV-class;
    queue 3 nc-CONTROL-class;
}
```

Related Topics

- Class of Service Using IPv6 Solutions Page
- Roadmap for Configuring CoS with IPv6 DiffServ on page 13
- Overview of Class of Service Using IPv6 DiffServ on page 3
- System Requirements for CoS with DiffServ for IPv6 on page 12
- Example: CoS with IPv6 DiffServ Configuration on page 24

Configuring Rewrite Rules

You configure rewrite rules to replace DSCPs on packets received from the customer with the values expected by other routers. Rewrite rules use the forwarding class information and packet loss priority (PLP) used internally by the router to establish the DSCP on outbound packets. To configure rewrite rules, include the rewrite-rules statement at the [edit class-of-service] hierarchy level.

```
[edit class-of-service]
rewrite-rules rewrite-IPv6-dscps {
    forwarding-class be-DATA-class {
        loss-priority low code points 000000;
        loss-priority high code points 000001;
    }
    forwarding-class ef-FIN-class {
        loss-priority low code points 101110;
```

```
            loss-priority high code points 101111;
        }
        forwarding-class af-AV-class {
            loss-priority low code points 001010;
            loss-priority high code points 001100;
        }
        forwarding-class nc-CONTROL-class {
            loss-priority low code points 110000;
            loss-priority high code points 110001;
        }
    }
```

Related Topics

- Class of Service Using IPv6 Solutions Page
- Applying Rewrite Rules to an Interface on page 17
- Roadmap for Configuring CoS with IPv6 DiffServ on page 13
- Overview of Class of Service Using IPv6 DiffServ on page 3
- System Requirements for CoS with DiffServ for IPv6 on page 12
- Example: CoS with IPv6 DiffServ Configuration on page 24

Applying Rewrite Rules to an Interface

To apply the configured rewrite rules, include the **rewrite-rules** statement at the [edit class-of-service interfaces] hierarchy level.

```
[edit class-of-service interfaces]
so-0/1/1 {
    unit 0 {
        rewrite-rules {
            dscp-ipv6 rewrite-IPv6-dscps;
        }
    }
}
```

Related Topics

- Class of Service Using IPv6 Solutions Page
- Configuring Rewrite Rules on page 16
- Roadmap for Configuring CoS with IPv6 DiffServ on page 13
- Overview of Class of Service Using IPv6 DiffServ on page 3
- System Requirements for CoS with DiffServ for IPv6 on page 12
- Example: CoS with IPv6 DiffServ Configuration on page 24

Configuring BA Classifiers

You configure BA classifiers for IPv6 on network interfaces because the DSCPs have been explicitly rewritten on the edge routers. To configure a BA classifier for IPv6 DSCPs, include the **dscp-ipv6** statement and give the classifier a name. Then import the default classifier and specify the forwarding class, loss priority, and code points for each established traffic class at the [edit class-of-service] hierarchy level.

```
[edit class-of-service]
classifiers {
   dscp-ipv6 IPv6-classifier {
      import default; # Uses the DSCP default map.
      forwarding-class be-DATA-class {
         loss-priority high code-points 000001;
      }
      forwarding-class ef-FIN-class {
         loss-priority high code-points 101111;
      }
      forwarding-class af-AV-class {
         loss-priority high code-points 001100;
      }
      forwarding-class nc-CONTROL-class {
         loss-priority high code-points 110001;
      }
   }
}
```

Related Topics

- Class of Service Using IPv6 Solutions Page
- Applying a BA Classifier to an Interface on page 18
- Roadmap for Configuring CoS with IPv6 DiffServ on page 13
- Overview of Class of Service Using IPv6 DiffServ on page 3
- System Requirements for CoS with DiffServ for IPv6 on page 12
- Example: CoS with IPv6 DiffServ Configuration on page 24

Applying a BA Classifier to an Interface

To apply the configured classifier, include the **classifiers** statement at the **[edit class-of-service interfaces]** hierarchy level.

```
[edit class-of-service interfaces]
so-0/1/1 {
   unit 0 {
      classifiers {
         dscp-ipv6 IPv6-classifier;
      }
   }
}
```

Related Topics

- Class of Service Using IPv6 Solutions Page
- Configuring BA Classifiers on page 17
- Roadmap for Configuring CoS with IPv6 DiffServ on page 13
- Overview of Class of Service Using IPv6 DiffServ on page 3
- System Requirements for CoS with DiffServ for IPv6 on page 12
- Example: CoS with IPv6 DiffServ Configuration on page 24

Configuring RED Drop Profiles

You configure RED drop profiles to determine the probability of DiffServ assured forwarding packets being discarded under congested conditions. To configure RED drop profiles for assured forwarding without the PLP bit set and with the PLP bit set, include the drop-profiles statement at the [edit class-of-service] hierarchy level.

```
[edit class-of-service]
drop-profiles {
    af-AV-normal {
        interpolate {
            fill-level [95 100];
            drop-probability [0 100];
        }
    }
    af-AV-with-PLP {
        interpolate {
            fill-level [60 70 80 90 95];
            drop-probability [80 90 95 97 100];
        }
    }
}
```

Assured forwarding traffic with the PLP bit set has a more aggressive drop probability than traffic without the PLP bit set.

Related Topics

- Class of Service Using IPv6 Solutions Page
- Roadmap for Configuring CoS with IPv6 DiffServ on page 13
- Overview of Class of Service Using IPv6 DiffServ on page 3
- System Requirements for CoS with DiffServ for IPv6 on page 12
- Example: CoS with IPv6 DiffServ Configuration on page 24

Configuring Schedulers

You configure schedulers to assign resources, priorities, and drop profiles to output queues. To configure a scheduler, include the schedulers statement at the [edit class-of-service] hierarchy level.

```
[edit class-of-service]
schedulers {
    be-DATA-scheduler {
        transmit-rate percent 40;
        buffer-size percent 40;
        priority low;
    }
    ef-FIN-scheduler {
        transmit-rate percent 10;
        buffer-size percent 10;
        priority high;
    }
```

```
    af-AV-scheduler {
        transmit-rate percent 45;
        buffer-size percent 45;
        priority high;
        drop-profile-map loss-priority low protocol any drop-profile af-AV-normal;
        drop-profile-map loss-priority high protocol any drop-profile af-AV-with-PLP;
    }
    nc-CONTROL-scheduler {
        transmit-rate percent 5;
        buffer-size percent 5;
        priority low;
    }
}
```

Related Topics

- Class of Service Using IPv6 Solutions Page
- Roadmap for Configuring CoS with IPv6 DiffServ on page 13
- Overview of Class of Service Using IPv6 DiffServ on page 3
- System Requirements for CoS with DiffServ for IPv6 on page 12
- Example: CoS with IPv6 DiffServ Configuration on page 24

Configuring Scheduler Maps

You configure a scheduler map to assign a forwarding class to a scheduler. To configure a scheduler map, include the **scheduler-maps** statement and scheduler name at the [edit class-of-service] hierarchy level.

```
[edit class-of-service]
scheduler-maps {
    diffserv-cos-map {
        forwarding-class be-DATA-class scheduler be-DATA-scheduler;
        forwarding-class ef-FIN-class scheduler ef-FIN-scheduler;
        forwarding-class af-AV-class scheduler af-AV-scheduler;
        forwarding-class nc-CONTROL-class scheduler nc-CONTROL-scheduler;
    }
}
```

Related Topics

- Class of Service Using IPv6 Solutions Page
- Applying a Scheduler Map to an Interface on page 20
- Roadmap for Configuring CoS with IPv6 DiffServ on page 13
- Overview of Class of Service Using IPv6 DiffServ on page 3
- System Requirements for CoS with DiffServ for IPv6 on page 12
- Example: CoS with IPv6 DiffServ Configuration on page 24

Applying a Scheduler Map to an Interface

To apply the configured scheduler map, include the **scheduler-map** statement at the [edit class-of-service] hierarchy level.

```
[edit class-of-service]
interfaces {
   so-1/0/1 {
      scheduler-map diffserv-cos-map;
   }
}
```

Related Topics

- Class of Service Using IPv6 Solutions Page
- Configuring Scheduler Maps on page 20
- Roadmap for Configuring CoS with IPv6 DiffServ on page 13
- Overview of Class of Service Using IPv6 DiffServ on page 3
- System Requirements for CoS with DiffServ for IPv6 on page 12
- Example: CoS with IPv6 DiffServ Configuration on page 24

Chapter 3

Class of Service with IPv6 DiffServ Example

Merging Examples

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following configuration to a file and name the file ex-script.conf. Copy the ex-script.conf file to the /var/tmp directory on your routing platform.

   ```
   system {
       scripts {
           commit {
               file ex-script.xsl;
           }
       }
   }
   interfaces {
       fxp0 {
           disable;
           unit 0 {
               family inet {
                   address 10.0.0.1/24;
               }
           }
       }
   }
   ```

2. Merge the contents of the file into your routing platform configuration by issuing the load merge configuration mode command:

   ```
   [edit]
   user@host# load merge /var/tmp/ex-script.conf
   ```

```
load complete
```

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following snippet to a file and name the file ex-script-snippet.conf. Copy the ex-script-snippet.conf file to the /var/tmp directory on your routing platform.

   ```
   commit {
     file ex-script-snippet.xsl; }
   ```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

   ```
   [edit]
   user@host# edit system scripts
   [edit system scripts]
   ```

3. Merge the contents of the file into your routing platform configuration by issuing the load merge relative configuration mode command:

   ```
   [edit system scripts]
   user@host# load merge relative /var/tmp/ex-script-snippet.conf
   load complete
   ```

Related Topics

- Example: CoS with IPv6 DiffServ Configuration on page 24
- Roadmap for Configuring CoS with IPv6 DiffServ on page 13
- Overview of Class of Service Using IPv6 DiffServ on page 3
- System Requirements for CoS with DiffServ for IPv6 on page 12

For more information about the load command, see the *JUNOS CLI User Guide*.

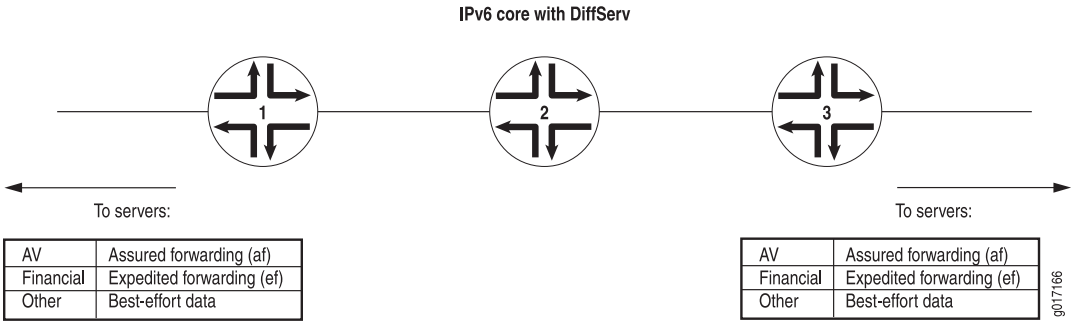
Example: CoS with IPv6 DiffServ Configuration

Example: CoS with IPv6 DiffServ Configuration

The example assigns expedited forwarding to Q1 and a subset of the assured forwarding classes (af1*x*) to Q2, and distributes resources among all four forwarding classes.

Figure 2 on page 25 shows the topology of the three routers and links that are used as a case study in this chapter.

Figure 2: Basic IPv6 DiffServ Topology



In this case study, the service provider has agreed to provide high-priority delivery of packets for two applications between the customer's servers at two sites. The first application generates streams of high-definition audiovisual (television) packet flows and the second generates large quantities of time-sensitive financial information. In all cases, the packet flow is from server to server. The service provider marks the packets appropriately as they enter the network from either site, configures special queues and forwarding classes for this traffic on the three routers, and uses DiffServ for IPv6 for this purpose.

Routers 1 and 3 use multifield (MF) classifiers on the customer-facing interfaces to detect high-priority packets and rewrite the Differentiated Services code points (DSCPs) appropriately. Best-effort data and network control packets are not affected. All three routers are configured with consistent schedulers and resources to handle high-priority packets properly.

Figure 3: IPv6 DiffServ Configuration

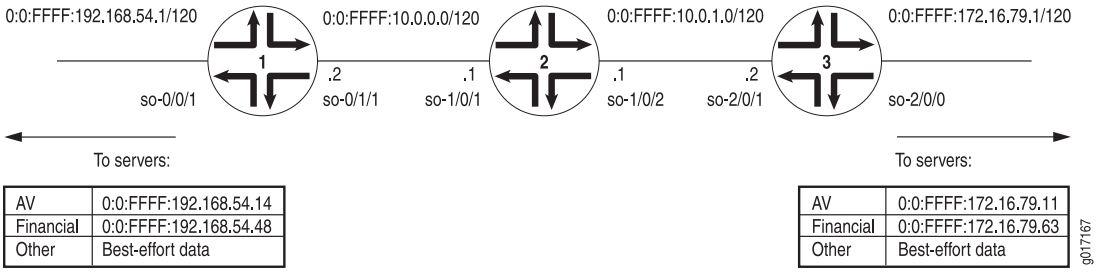


Figure 3 on page 25 shows the complete topology for IPv6 DiffServ, complete with interfaces and IPv6 addresses. The IPv4-mapped IPv6 address format described in RFC 1884 is used.

Begin your configuration on Router 2, the core router. This ensures that when DiffServ is enabled on the edge routers, class of service (CoS) is enabled end to end through the network. The core router configuration is a little simpler because no MF classification is configured in the core.

Router 2

```
[edit]
class-of-service {
   classifiers { # Router 2 classifiers.
      dscp-ipv6 IPv6-classifier {
         import default; # Uses the DSCP default map.
```

```
        forwarding-class be-DATA-class {
            loss-priority high code-points 000001;
        }
        forwarding-class ef-FIN-class {
            loss-priority high code-points 101111;
        }
        forwarding-class af-AV-class {
            loss-priority high code-points 001100;
        }
        forwarding-class nc-CONTROL-class {
            loss-priority high code-points 110001;
        }
    }
}
drop-profiles { # Router 2 drop profiles.
    af-AV-normal {
        interpolate {
            fill-level [95 100];
            drop-probability [0 100];
        }
    }
    af-AV-with-PLP {
        interpolate {
            fill-level [60 70 80 90 95];
            drop-probability [80 90 95 97 100];
        }
    }
}
forwarding-classes { # Router 2 forwarding classes.
    queue 0 be-DATA-class;
    queue 1 ef-FIN-class;
    queue 2 af-AV-class;
    queue 3 nc-CONTROL-class;
}
interfaces { # Router 2 class-of-service interfaces.
    so-1/0/1 { # Connected to R1.
        scheduler-map diffserv-cos-map;
        unit 0 {
            classifiers {
                dscp-ipv6 IPv6-classifier;
            }
            rewrite-rules {
                dscp-ipv6 rewrite-IPv6-dscp;
            }
        }
    }
    so-1/0/2 { # Connected to R3.
        scheduler-map diffserv-cos-map;
        unit 0 {
            classifiers {
                dscp-ipv6 IPv6-classifier;
            }
            rewrite-rules {
                dscp-ipv6 rewrite-IPv6-dscp;
            }
        }
```

```
        }
    }
    rewrite-rules rewrite-IPv6-dscps { # Router 2 rewrite rules.
        forwarding-class be-DATA-class {
            loss-priority low code points 000000;
            loss-priority high code points 000001;
        }
        forwarding-class ef-FIN-class {
            loss-priority low code points 101110;
            loss-priority high code points 101111;
        }
        forwarding-class af-AV-class {
            loss-priority low code points 001010;
            loss-priority high code points 001100;
        }
        forwarding-class nc-CONTROL-class {
            loss-priority low code points 110000;
            loss-priority high code points 110001;
        }
    }
    scheduler-maps { # Router 2 scheduler maps.
        diffserv-cos-map {
            forwarding-class be-DATA-class scheduler be-DATA-scheduler;
            forwarding-class ef-FIN-class scheduler ef-FIN-scheduler;
            forwarding-class af-AV-class scheduler af-AV-scheduler;
            forwarding-class nc-CONTROL-class scheduler nc-CONTROL-scheduler;
        }
    }
    schedulers { # Router 2 schedulers.
        be-DATA-scheduler {
            transmit-rate percent 40;
            buffer-size percent 40;
            priority low;
        }
        ef-FIN-scheduler {
            transmit-rate percent 10;
            buffer-size percent 10;
            priority high;
        }
        af-AV-scheduler {
            transmit-rate percent 45;
            buffer-size percent 45;
            priority high;
            drop-profile-map loss-priority low protocol any drop-profile af-AV-normal;
            drop-profile-map loss-priority high protocol any drop-profile af-AV-with-PLP;
        }
        nc-CONTROL-scheduler {
            transmit-rate percent 5;
            buffer-size percent 5;
            priority low;
        }
    }
}
interfaces { # R2 interfaces.
    so-1/0/1 { # Connected to R1.
        unit 0 {
```

```
            family inet {
               address 10.0.0.1/24;
            }
            family inet6 {
               address 0:0:FFFF:10.0.0.1/120;
            }
         }
      }
      so-1/0/2 { # Connected to R3.
         unit 0 {
            family inet {
               address 10.0.1.1/24;
            }
            family inet6 {
               address 0:0:FFFF:10.0.1.1/120;
            }
         }
      }
   }
```

Continue your configuration on Router 1 and Router 3, the edge routers. These routers get firewall-filter-based MF classifiers and rewrite rules for markers as well as schedulers and drop profiles on the core-facing interfaces.

Router 1

```
[edit]
class-of-service {
   classifiers { # Router 1 classifiers.
      dscp-ipv6 IPv6-classifier {
         import default; # Uses the DSCP default map.
         forwarding-class be-DATA-class {
            loss-priority high code-points 000001;
         }
         forwarding-class ef-FIN-class {
            loss-priority high code-points 101111;
         }
         forwarding-class af-AV-class {
            loss-priority high code-points 001100;
         }
         forwarding-class nc-CONTROL-class {
            loss-priority high code-points 110001;
         }
      }
   }
   drop-profiles { # Router 1 drop profiles.
      af-AV-normal {
         interpolate {
            fill-level [95 100];
            drop-probability [0 100];
         }
      }
      af-AV-with-PLP {
         interpolate {
            fill-level [60 70 80 90 95];
            drop-probability [80 90 95 97 100];
         }
      }
```

```
}
forwarding-classes { # Router 1 forwarding classes.
   queue 0 be-DATA-class;
   queue 1 ef-FIN-class;
   queue 2 af-AV-class;
   queue 3 nc-CONTROL-class;
}
interfaces { # Router 1 class-of-service interfaces.
   so-0/1/1 { # To servers.
      scheduler-map diffserv-cos-map;
      unit 0 {
         classifiers {
            dscp-ipv6 IPv6-classifier;
         }
         rewrite-rules {
            dscp-ipv6 rewrite-IPv6-dscp;
         }
      }
   }
   rewrite-rules rewrite-IPv6-dscps { # Router 1 rewrite rules.
      forwarding-class be-DATA-class {
         loss-priority low code points 000000;
         loss-priority high code points 000001;
      }
      forwarding-class ef-FIN-class {
         loss-priority low code points 101110;
         loss-priority high code points 101111;
      }
      forwarding-class af-AV-class {
         loss-priority low code points 001010;
         loss-priority high code points 001100;
      }
      forwarding-class nc-CONTROL-class {
         loss-priority low code points 110000;
         loss-priority high code points 110001;
      }
   }
   scheduler-maps { # Router 1 scheduler map.
      diffserv-cos-map {
         forwarding-class be-DATA-class scheduler be-DATA-scheduler;
         forwarding-class ef-FIN-class scheduler ef-FIN-scheduler;
         forwarding-class af-AV-class scheduler af-AV-scheduler;
         forwarding-class nc-CONTROL-class scheduler nc-CONTROL-scheduler;
      }
   }
   schedulers { # Router 1 schedulers.
      be-DATA-scheduler {
         transmit-rate percent 40;
         buffer-size percent 40;
         priority low;
      }
      ef-FIN-scheduler {
         transmit-rate percent 10;
         buffer-size percent 10;
         priority high;
      }
```

```
        af-AV-scheduler {
            transmit-rate percent 45;
            buffer-size percent 45;
            priority high;
            drop-profile-map loss-priority low protocol any drop-profile af-AV-normal;
            drop-profile-map loss-priority high protocol any drop-profile af-AV-with-PLP;
        }
        nc-CONTROL-scheduler {
            transmit-rate percent 5;
            buffer-size percent 5;
            priority low;
        }
    }
}
firewall { # Router 1 firewall policer and filter.
    policer ef-FIN-Policer-Profile {
        if-exceeding {
            bandwidth-percent 10;
            burst-size-limit 2k;
        }
        then loss-priority high;
    }
    family inet6 {
        filter mf-classifier {
            filter-specific;
            term AV {
                from {
                    destination-address {
                        0:0:FFFF:172.16.79.11;
                    }
                }
                then {
                    loss-priority low;
                    forwarding-class af-AV-class;
                }
            }
            term Finance {
                from {
                    destination-address {
                        0:0:FFFF:172.16.79.63;
                    }
                }
                then {
                    policer ef-FIN-Policer-Profile;
                    forwarding-class ef-FIN-class;
                }
            }
            term Network-Control {
                from {
                    traffic-class 192; # 192 is the 110000 traffic class.
                }
                then {
                    forwarding-class nc-CONTROL-class; # This is network control traffic.
                }
            }
            term Data {
```

```
            then forwarding-class be-DATA-class; # The rest is data.
          }
        }
      }
    }
    interfaces { # Router 1 interfaces.
      so-0/0/1 { # To servers.
        unit 0 {
          family inet {
            address 192.168.54.1/24;
          }
          family inet6 {
            filter {
              input mf-classifier;
            }
            address 0:0:FFFF:192.168.54.1/120;
          }
        }
      }
      so-0/1/1 { # Connected to R2.
        unit 0 {
          family inet {
            address 10.0.0.2/24;
          }
          family inet6 {
            address 0:0:FFFF:10.0.0.2/120;
          }
        }
      }
    }
}
```

Router 3

```
[edit]
class-of-service {
  classifiers { # Router 3 classifiers.
    dscp-ipv6 IPv6-classifier {
      import default; # Uses the DSCP default map.
      forwarding-class be-DATA-class {
        loss-priority high code-points 000001;
      }
      forwarding-class ef-FIN-class {
        loss-priority high code-points 101111;
      }
      forwarding-class af-AV-class {
        loss-priority high code-points 001100;
      }
      forwarding-class nc-CONTROL-class {
        loss-priority high code-points 110001;
      }
    }
  }
  drop-profiles { # Router 3 drop profiles.
    af-AV-normal {
      interpolate {
        fill-level [95 100];
```

```
            drop-probability [0 100];
        }
    }
    af-AV-with-PLP {
        interpolate {
            fill-level [60 70 80 90 95];
            drop-probability [80 90 95 97 100];
        }
    }
}
forwarding-classes { # Router 3 forwarding classes.
    queue 0 be-DATA-class;
    queue 1 ef-FIN-class;
    queue 2 af-AV-class;
    queue 3 nc-CONTROL-class;
}
interfaces { # Router 3 class-of-service interfaces.
    so-2/0/1 { # To servers.
        scheduler-map diffserv-cos-map;
        unit 0 {
            classifiers {
                dscp-ipv6 IPv6-classifier;
            }
            rewrite-rules {
                dscp-ipv6 rewrite-IPv6-dscp;
            }
        }
    }
    rewrite-rules rewrite-IPv6-dscps { # Router 3 rewrite rules.
        forwarding-class be-DATA-class {
            loss-priority low code points 000000;
            loss-priority high code points 000001;
        }
        forwarding-class ef-FIN-class {
            loss-priority low code points 101110;
            loss-priority high code points 101111;
        }
        forwarding-class af-AV-class {
            loss-priority low code points 001010;
            loss-priority high code points 001100;
        }
        forwarding-class nc-CONTROL-class {
            loss-priority low code points 110000;
            loss-priority high code points 110001;
        }
    }
    scheduler-maps { # Router 3 scheduler map.
        diffserv-cos-map {
            forwarding-class be-DATA-class scheduler be-DATA-scheduler;
            forwarding-class ef-FIN-class scheduler ef-FIN-scheduler;
            forwarding-class af-AV-class scheduler af-AV-scheduler;
            forwarding-class nc-CONTROL-class scheduler nc-CONTROL-scheduler;
        }
    }
    schedulers { # Router 3 schedulers.
        be-DATA-scheduler {
```

```
        transmit-rate percent 40;
        buffer-size percent 40;
        priority low;
    }
    ef-FIN-scheduler {
        transmit-rate percent 10;
        buffer-size percent 10;
        priority high;
    }
    af-AV-scheduler {
        transmit-rate percent 45;
        buffer-size percent 45;
        priority high;
        drop-profile-map loss-priority low protocol any drop-profile af-AV-normal;
        drop-profile-map loss-priority high protocol any drop-profile af-AV-with-PLP;
    }
    nc-CONTROL-scheduler {
        transmit-rate percent 5;
        buffer-size percent 5;
        priority low;
    }
}
firewall { # Router 3 firewall policer and filter.
    policer ef-FIN-Policer-Profile {
        if-exceeding {
            bandwidth-percent 10;
            burst-size-limit 2k;
        }
        then loss-priority high;
    }
    family inet6 {
        filter mf-classifier {
            filter-specific;
            term AV {
                from {
                    destination-address {
                        0:0:FFFF:172.16.79.11;
                    }
                }
                then {
                    loss-priority low;
                    forwarding-class af-AV-class;
                }
            }
            term Finance {
                from {
                    destination-address {
                        0:0:FFFF:172.16.79.63;
                    }
                }
                then {
                    policer ef-FIN-Policer-Profile;
                    forwarding-class ef-FIN-class;
                }
            }
            term Network-Control {
```

```
                from {
                    traffic-class 192; # 192 is the 110000 traffic class.
                }
                then {
                    forwarding-class nc-CONTROL-class; # This is network control traffic.
                }
            }
            term Data {
                then forwarding-class be-DATA-class; # The rest is data.
            }
        }
    }
}
interfaces { # Router 3 interfaces.
    so-2/0/0 { # To servers.
        unit 0 {
            family inet {
                address 1172.16.79.1/24;
            }
            family inet6 {
                filter {
                    input mf-classifier;
                }
                address 0:0:FFFF:172.16.79.1/120;
            }
        }
    }
    so-2/0/1 { # to R2
        unit 0 {
            family inet {
                address 10.0.1.2/24;
            }
            family inet6 {
                address 0:0:FFFF:10.0.1.2/120;
            }
        }
    }
}
}
```

Verifying Your Work

To verify that your CoS using IPv6 DiffServ configuration is correct, use the following commands:

- show class-of-service classifier type dscp-ipv6
- show class-of-service rewrite-rule type dscp-ipv6
- show class-of-service interface
- show class-of-service forwarding-table classifier mapping
- show class-of-service forwarding-table rewrite-rule mapping

- show class-of-service scheduler-map *scheduler-map-name*
- show class-of-service forwarding-table scheduler-map

The following section shows the output of these commands used with the configuration example.

DiffServ Classifiers

```
user@R1> show class-of-service classifier type dscp-ipv6
Classifier: dscp-ipv6-default, Code point type: dscp-ipv6, Index: 4
  Code point         Forwarding class                  Loss priority
  000000             be-DATA-class                     low
  000001             be-DATA-class                     low
  000010             be-DATA-class                     low
  000011             be-DATA-class                     low
  000100             be-DATA-class                     low
  000101             be-DATA-class                     low
  000110             be-DATA-class                     low
  000111             be-DATA-class                     low
  001000             be-DATA-class                     low
  001001             be-DATA-class                     low
  001010             af-AV-class                       low
  001011             be-DATA-class                     low
  001100             af-AV-class                       high
  001101             be-DATA-class                     low
  001110             af-AV-class                       high
  001111             be-DATA-class                     low
  010000             be-DATA-class                     low
  010001             be-DATA-class                     low
  010010             be-DATA-class                     low
  010011             be-DATA-class                     low
  010100             be-DATA-class                     low
  010101             be-DATA-class                     low
  010110             be-DATA-class                     low
  010111             be-DATA-class                     low
  011000             be-DATA-class                     low
  011001             be-DATA-class                     low
  011010             be-DATA-class                     low
  011011             be-DATA-class                     low
  011100             be-DATA-class                     low
  011101             be-DATA-class                     low
  011110             be-DATA-class                     low
  011111             be-DATA-class                     low
  100000             be-DATA-class                     low
  100001             be-DATA-class                     low
  100010             be-DATA-class                     low
  100011             be-DATA-class                     low
  100100             be-DATA-class                     low
  100101             be-DATA-class                     low
  100110             be-DATA-class                     low
  100111             be-DATA-class                     low
  101000             be-DATA-class                     low
  101001             be-DATA-class                     low
  101010             be-DATA-class                     low
  101011             be-DATA-class                     low
  101100             be-DATA-class                     low
  101101             be-DATA-class                     low
  101110             ef-FIN-class                      low
  101111             be-DATA-class                     low
  110000             nc-CONTROL-class                  low
  110001             be-DATA-class                     low
```

```
  110010              be-DATA-class                       low
  110011              be-DATA-class                       low
  110100              be-DATA-class                       low
  110101              be-DATA-class                       low
  110110              be-DATA-class                       low
  110111              be-DATA-class                       low
  111000              nc-CONTROL-class                    low
  111001              be-DATA-class                       low
  111010              be-DATA-class                       low
  111011              be-DATA-class                       low
  111100              be-DATA-class                       low
  111101              be-DATA-class                       low
  111110              be-DATA-class                       low
  111111              be-DATA-class                       low
Classifier: IPv6-classifier, Code point type: dscp-ipv6, Index: 18301
  Code point          Forwarding class                    Loss priority
  000000              be-DATA-class                       low
  000001              be-DATA-class                       high
  000010              be-DATA-class                       low
  000011              be-DATA-class                       low
  000100              be-DATA-class                       low
  000101              be-DATA-class                       low
  000110              be-DATA-class                       low
  000111              be-DATA-class                       low
  001000              be-DATA-class                       low
  001001              be-DATA-class                       low
  001010              af-AV-class                         low
  001011              be-DATA-class                       low
  001100              af-AV-class                         high
  001101              be-DATA-class                       low
  001110              af-AV-class                         high
  001111              be-DATA-class                       low
  010000              be-DATA-class                       low
  010001              be-DATA-class                       low
  010010              be-DATA-class                       low
  010011              be-DATA-class                       low
  010100              be-DATA-class                       low
  010101              be-DATA-class                       low
  010110              be-DATA-class                       low
  010111              be-DATA-class                       low
  011000              be-DATA-class                       low
  011001              be-DATA-class                       low
  011010              be-DATA-class                       low
  011011              be-DATA-class                       low
  011100              be-DATA-class                       low
  011101              be-DATA-class                       low
  011110              be-DATA-class                       low
  011111              be-DATA-class                       low
  100000              be-DATA-class                       low
  100001              be-DATA-class                       low
  100010              be-DATA-class                       low
  100011              be-DATA-class                       low
  100100              be-DATA-class                       low
  100101              be-DATA-class                       low
  100110              be-DATA-class                       low
  100111              be-DATA-class                       low
  101000              be-DATA-class                       low
  101001              be-DATA-class                       low
  101010              be-DATA-class                       low
  101011              be-DATA-class                       low
  101100              be-DATA-class                       low
```

```
  101101            be-DATA-class                 low
  101110            ef-FIN-class                  low
  101111            ef-FIN-class                  high
  110000            nc-CONTROL-class              low
  110001            nc-CONTROL-class              high
  110010            be-DATA-class                 low
  110011            be-DATA-class                 low
  110100            be-DATA-class                 low
  110101            be-DATA-class                 low
  110110            be-DATA-class                 low
  110111            be-DATA-class                 low
  111000            nc-CONTROL-class              low
  111001            be-DATA-class                 low
  111010            be-DATA-class                 low
  111011            be-DATA-class                 low
  111100            be-DATA-class                 low
  111101            be-DATA-class                 low
  111110            be-DATA-class                 low
  111111            be-DATA-class                 low
```

Rewrite Rules

```
user@R1> show class-of-service rewrite-rule type dscp-ipv6
Rewrite rule: dscp-ipv6-default, Code point type: dscp-ipv6, Index: 20
  Forwarding class                Loss priority       Code point
  be-DATA-class                   low                 000000
  be-DATA-class                   high                000000
  ef-FIN-class                    low                 101110
  ef-FIN-class                    high                101110
  af-AV-class                     low                 001010
  af-AV-class                     high                001100
  nc-CONTROL-class                low                 110000
  nc-CONTROL-class                high                111000
Rewrite rule: rewrite-IPv6-dscp, Code point type: dscp-ipv6, Index: 58077
  Forwarding class                Loss priority       Code point
  be-DATA-class                   low                 000000
  be-DATA-class                   high                000001
  ef-FIN-class                    low                 101110
  ef-FIN-class                    high                101111
  af-AV-class                     low                 001010
  af-AV-class                     high                001100
  nc-CONTROL-class                low                 110000
  nc-CONTROL-class                high                110001
```

Class-of-Service Interfaces

```
user@R1> show class-of-service interface
...
Physical interface: so-0/0/1, Index: 141
Queues supported: 4, Queues in use: 4
  Scheduler map: diffserv-cos-map, Index: -543019056
  Logical interface: so-0/0/1.0, Index: 68
    Object         Name                  Type                      Index
    Rewrite        rewrite-IPv6-dscp     dscp-ipv6                 58077
    Rewrite        exp-default           exp                          21
    Classifier     IPv6-classifier       dscp-ipv6                 18301
    Classifier     exp-default           exp                           5
...
Physical interface: so-0/1/1, Index: 144
Queues supported: 4, Queues in use: 4
  Scheduler map: <default>, Index: -113795564

  Logical interface: so-0/1/1.0, Index: 69
    Object         Name                  Type                      Index
    Rewrite        exp-default           exp                          21
```

```
    Classifier     exp-default            exp                            5
    Classifier     ipprec-compatibility   ip                             8
```

Classifier Mapping

```
user@R1> show class-of-service forwarding-table classifier mapping
                        Table Index/
Interface      Index      Q num      Table type
so-0/0/1.0        68      18301      IPv6 DSCP
so-0/1/1.0        69          8      IPv4 precedence
```

Rewrite Rule Mapping

```
user@R1> show class-of-service forwarding-table rewrite-rule mapping
Interface      Index   Table index  Type
so-0/1/1.0        68      58077     IPv6 DSCP
```

Scheduler Map

```
user@R1> show class-of-service scheduler-map diffserv-cos-map
Scheduler map: diffserv-cos-map, Index: 1094596010
  Scheduler: be-DATA-scheduler, Forwarding class: be-DATA-class, Index: 14343
    Transmit rate: 40 percent, Rate Limit: none, Buffer size: 40 percent,
    Priority: low
    Drop profiles:
      Loss priority   Protocol    Index    Name
      Low             non-TCP         1    <default-drop-profile>
      Low             TCP             1    <default-drop-profile>
      High            non-TCP         1    <default-drop-profile>
      High            TCP             1    <default-drop-profile>
  Scheduler: ef-FIN-scheduler, Forwarding class: ef-FIN-class, Index: 21707
    Transmit rate: 10 percent, Rate Limit: none, Buffer size: 10 percent,
    Priority: high
    Drop profiles:
      Loss priority   Protocol    Index    Name
      Low             non-TCP         1    <default-drop-profile>
      Low             TCP             1    <default-drop-profile>
      High            non-TCP         1    <default-drop-profile>
      High            TCP             1    <default-drop-profile>
  Scheduler: af-AV-scheduler, Forwarding class: af-AV-class, Index: 51704
    Transmit rate: 45 percent, Rate Limit: none, Buffer size: 45 percent,
    Priority: high
    Drop profiles:
      Loss priority   Protocol    Index    Name
      Low             non-TCP     61474    af-AV-normal
      Low             TCP         61474    af-AV-normal
      High            non-TCP     65199    af-AV-with-PLP
      High            TCP         65199    af-AV-with-PLP
  Scheduler: nc-CONTROL-scheduler, Forwarding class: nc-CONTROL-class, Index:
50404
    Transmit rate: 5 percent, Rate Limit: none, Buffer size: 5 percent,
    Priority: low
    Drop profiles:
      Loss priority   Protocol    Index    Name
      Low             non-TCP         1    <default-drop-profile>
      Low             TCP             1    <default-drop-profile>
      High            non-TCP         1    <default-drop-profile>
      High            TCP             1    <default-drop-profile>
user@R1> show class-of-service forwarding-table scheduler-map
...
Interface: so-0/0/1 (Index: 141, Map index: -543019056, Map type: FINAL,
Num of queues: 4):
  Entry 0 (Scheduler index: 14343, Queue #: 0):
    Tx rate: 0 Kb (40%), Buffer size: 40 percent
Priority low
    PLP high: 1, PLP low: 1, TCP PLP high: 1, TCP PLP low: 1
```

```
  Entry 1 (Scheduler index: 21707, Queue #: 1):
    Tx rate: 0 Kb (10%), Buffer size: 10 percent
Priority high
    PLP high: 1, PLP low: 1, TCP PLP high: 1, TCP PLP low: 1
  Entry 2 (Scheduler index: 51704, Queue #: 2):
    Tx rate: 0 Kb (45%), Buffer size: 45 percent
Priority high
    PLP high: 65199, PLP low: 61474, TCP PLP high: 65199, TCP PLP low: 61474
  Entry 3 (Scheduler index: 50404, Queue #: 3):
    Tx rate: 0 Kb (5%), Buffer size: 5 percent
Priority low
    PLP high: 1, PLP low: 1, TCP PLP high: 1, TCP PLP low: 1
...
```

Related Topics

- Class of Service Using IPv6 Solutions Page
- Overview of Class of Service Using IPv6 DiffServ on page 3
- System Requirements for CoS with DiffServ for IPv6 on page 12
- Roadmap for Configuring CoS with IPv6 DiffServ on page 13

For More Information about CoS using DiffServ and IPv6

For additional information about CoS using DiffServ and IPv6, see the following:

- RFC 1924, *A Compact Representation of IPv6 Addresses*
- RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*
- RFC 2475, *An Architecture for Differentiated Services*
- RFC 2640, *Internet Protocol, Version 6 (IPv6) Specification*
- RFC 2983, *Differentiated Service and Tunnels*
- RFC 3260, *New Terminology and Clarifications for DiffServ*
- RFC 3317, *Differentiated Services Quality of Service Policy Information Base*
- RFC 3513, *IP Version 6 Addressing Architecture*

Related Topics

- Class of Service Using IPv6 Solutions Page
- Overview of Class of Service Using IPv6 DiffServ on page 3
- System Requirements for CoS with DiffServ for IPv6 on page 12
- Roadmap for Configuring CoS with IPv6 DiffServ on page 13

Part 2
Index

Index